



Step by Step: Extending SAP Access Control with SAP Fiori Applications

Vyacheslav (Slava) Plyushchikov
Advanced View Computer Technologies

Produced by Wellesley Information Services, LLC, publisher of SAPinsider. © 2016 Wellesley Information Services. All rights reserved.



In This Session

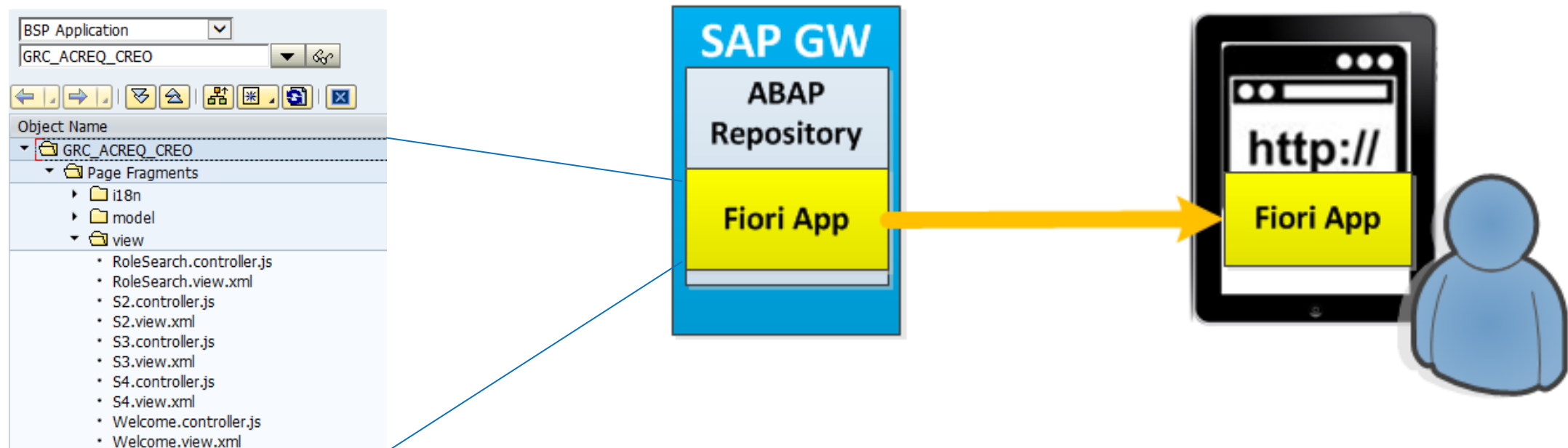
- Step-by-step process of:
 - ♦ Planning Fiori apps enhancements
 - ♦ Deploying Web IDE
 - ♦ Extending GRC Access Control Fiori apps
 - ♦ Tracing and debugging Fiori app front end
 - ♦ Testing and debugging OData calls
 - ♦ Tips and tricks of deploying extended Fiori apps

What We'll Cover

- **GRC Access Control Fiori overview**
- **Business scenario for Fiori app extension**
- **Implementation: IDE setup**
- **Implementation: Fiori extension project**
- **Tracing and debugging**
- **Deployment**
- **Tips and tricks**
- **Wrap-up**

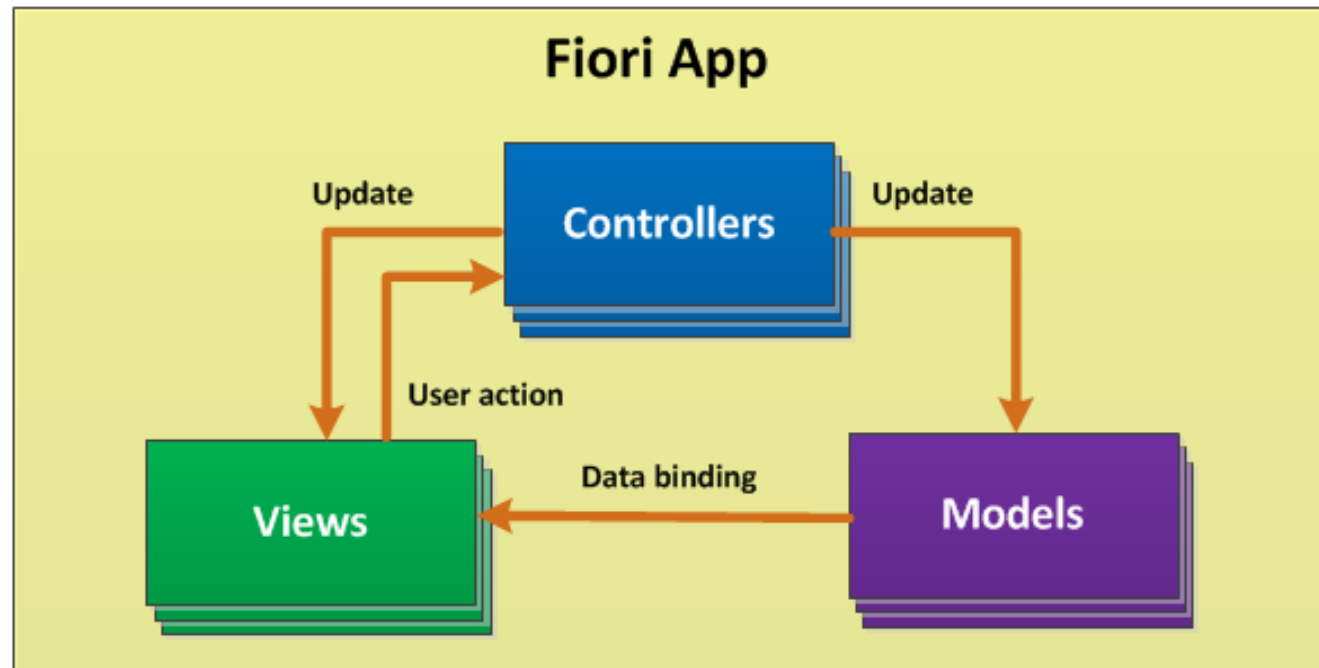
SAP Fiori: Technology

- Technology is based on HTML5/SAPUI5
- Fiori app resides in the ABAP repository as BSP application
- When executed, entire Fiori app is transferred to the HTML5-compatible web browser



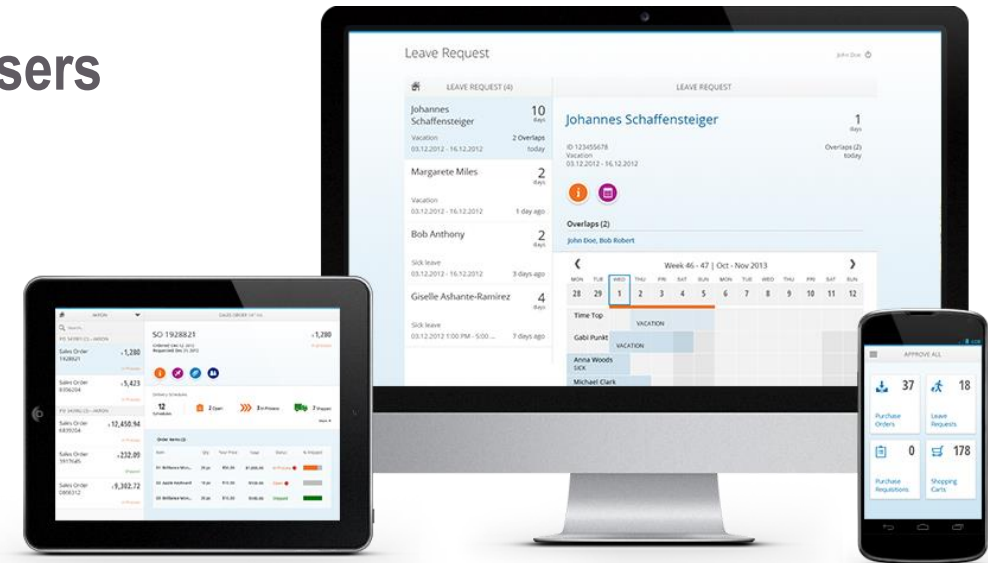
SAP Fiori: Technology (cont.)

- Model-View-Controller (MVC) concept
 - ♦ Separation between data presentation (view), flow logic (controller), and data persistence (model)
 - ♦ Better code reuse: change to one component does not affect other components



SAP Fiori: Benefits

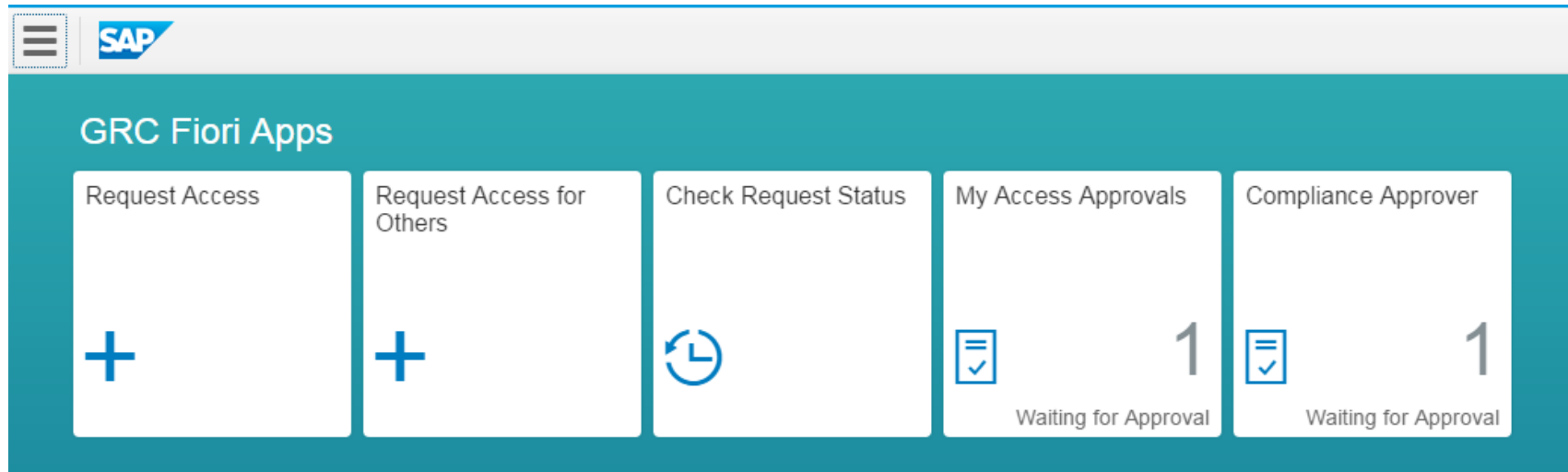
- Supported across multiple platforms, including mobile devices
 - ♦ Any device with HTML5-compatible browser can be used
- Requires zero footprint on the user device
 - ♦ No application deployment: entire app is loaded to the browser during the call
- Easy maintenance and support
 - ♦ Update on the server immediately available to all users



Source: SAP

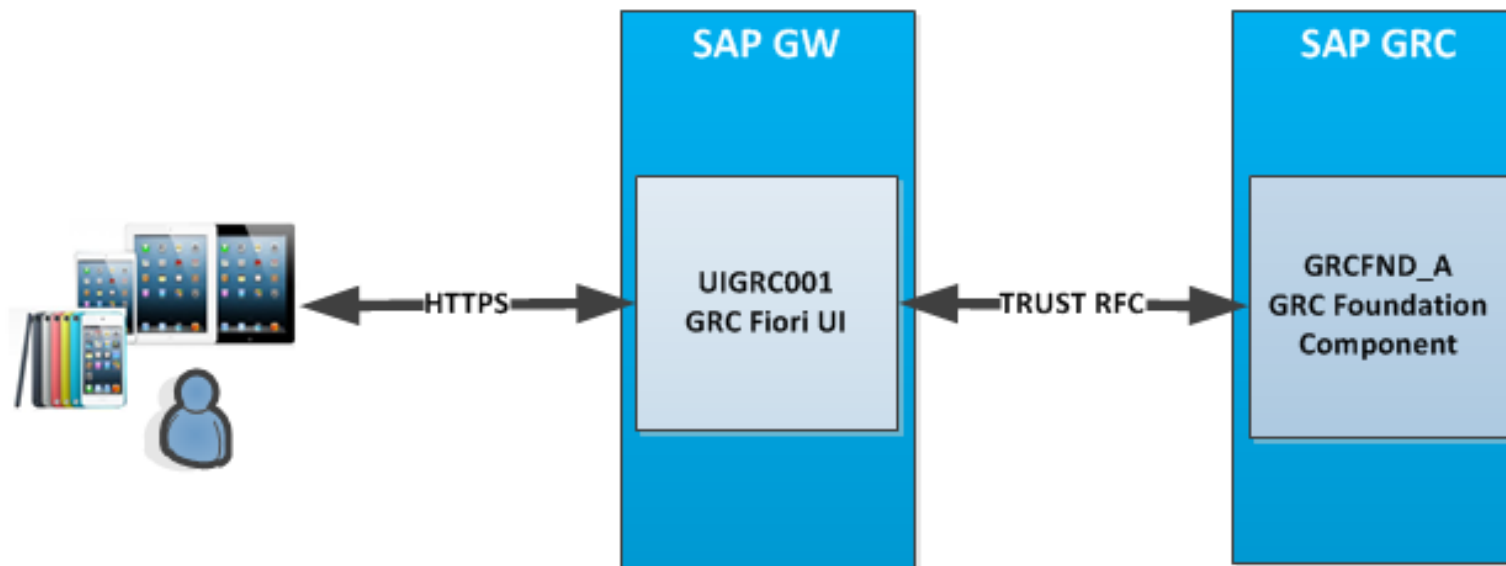
SAP GRC AC Fiori Suite

- Out-of-the-box transactional Fiori apps:
 - ♦ Request Access
 - ♦ Request Access for Others
 - ♦ Check Request Status
 - ♦ My Access Approvals
 - ♦ Compliance Approver



SAP GRC AC Fiori Suite (cont.)

- Front-end Component UIGRC001 is deployed on SAP Gateway
- Back-end OData services delivered in GRCFND_A on SAP GRC Server

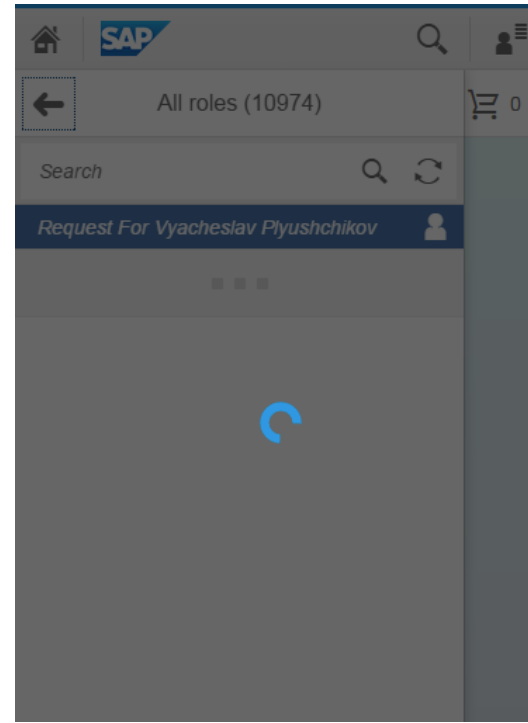
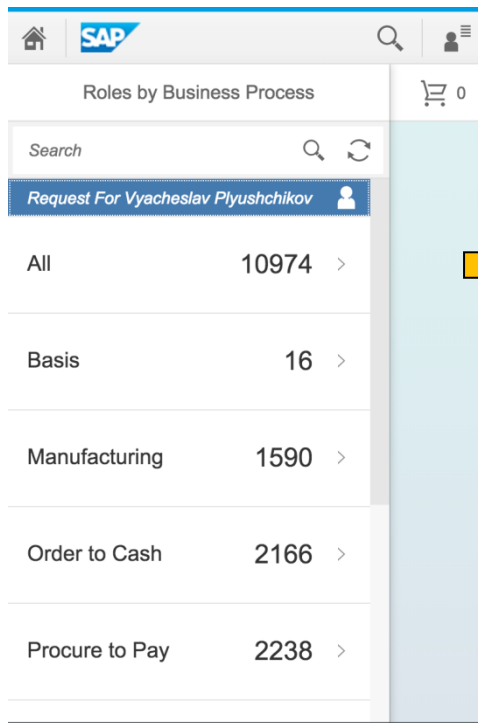


What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Business Scenario for Fiori Extension

- Issue with access request creation:
 - ♦ Company Role Catalog does not fit into out-of-the-box Fiori app functionality
 - ▶ Users cannot efficiently select roles
 - ▶ Performance issue due to large number of entries returned



Large number of roles, long wait time

Business Scenario for Fiori Extension (cont.)

- Standard app allows search only by business process or role keyword
- Company role catalog is based on employee geographical location and job function

Business Process	Role Count
PTP	2K roles
OTC	3K roles
RTR	...K roles
...	...

Each Business processes many roles. It is difficult to find the correct one.

Location	Job	Role Count
Argentina	Affiliate	5 roles
	Logistics	10 roles
	Tax	7 roles
	Treasury	5 roles

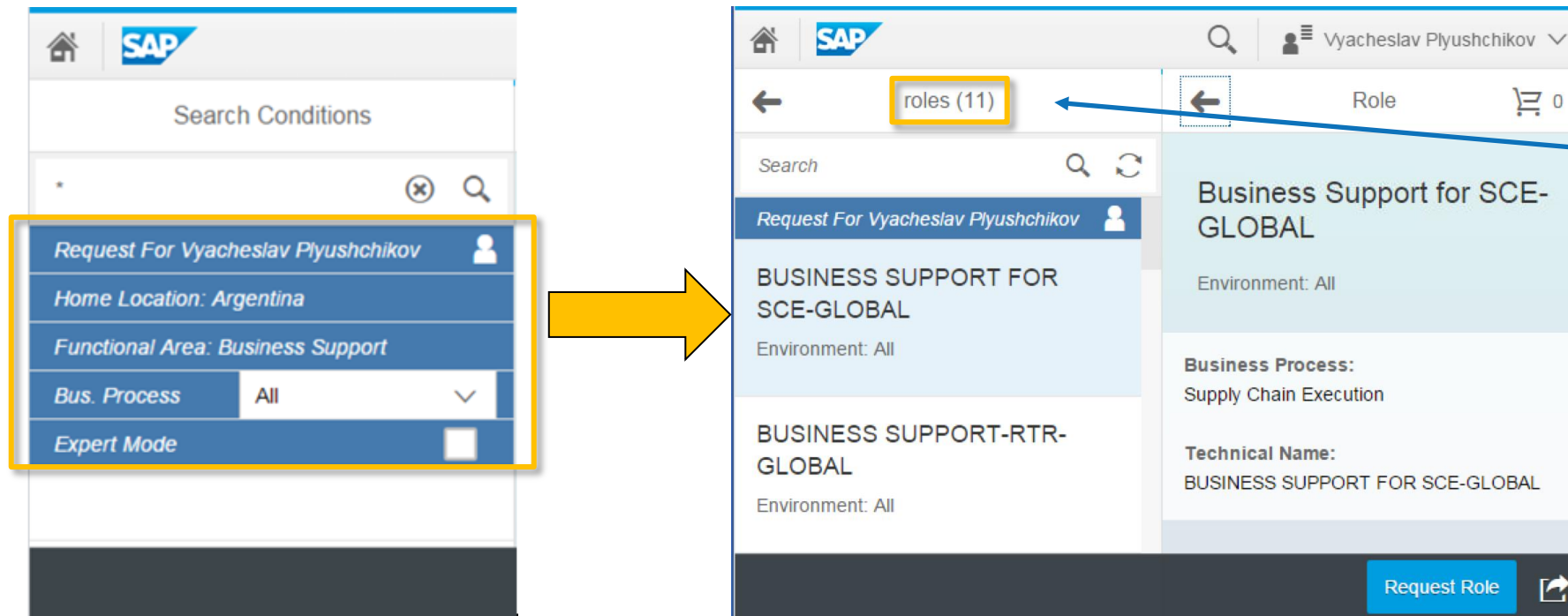
Brazil	Accounting	3 roles
	Affiliate	5 roles
	Logistics	3 roles
	Tax	7 roles

...
...

With additional filtering by Location and Job, number of roles to choose becomes reasonable

Business Scenario for Fiori Extension (cont.)

- Solution: Extend “Request Access for Others” app
 - ♦ Add required search fields
 - ♦ Allow “Expert” mode for Super Users



Now only 11 roles are returned. Easy for user to choose from.

Demo: Extended Fiori App

Extended Fiori App in Action: Request Access for Other



Demo

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

IDE Setup

- Option 1: Cloud/SAP HCP
 - ♦ SAP Web IDE hosted on SAP HCP
 - ♦ Free trial option and subscription-based option
- Option 2: Local Web IDE installation
 - ♦ Installed locally on Windows Workstation
 - ♦ Free trial version of Web IDE 1.12

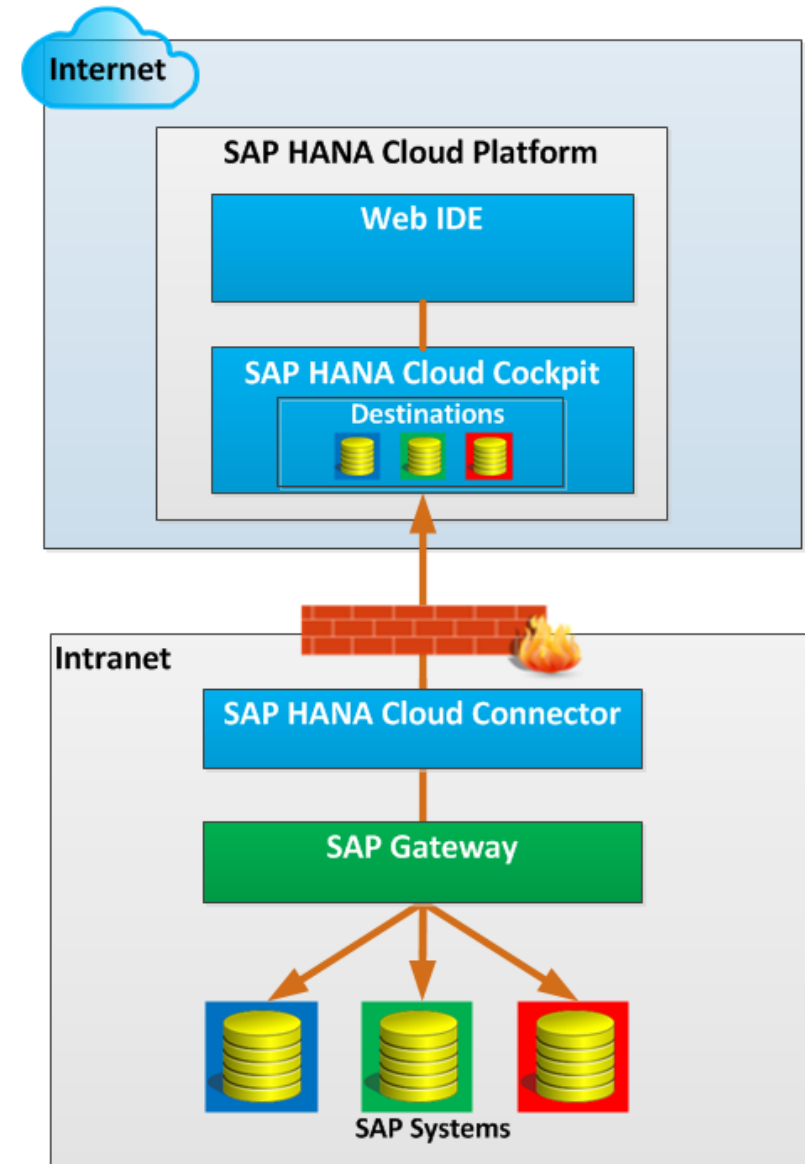


VS.



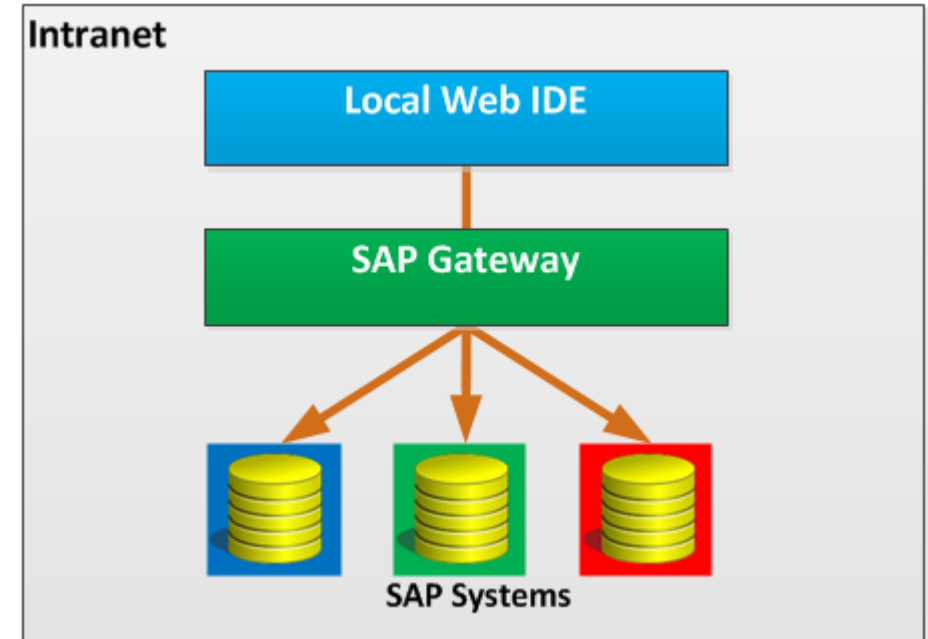
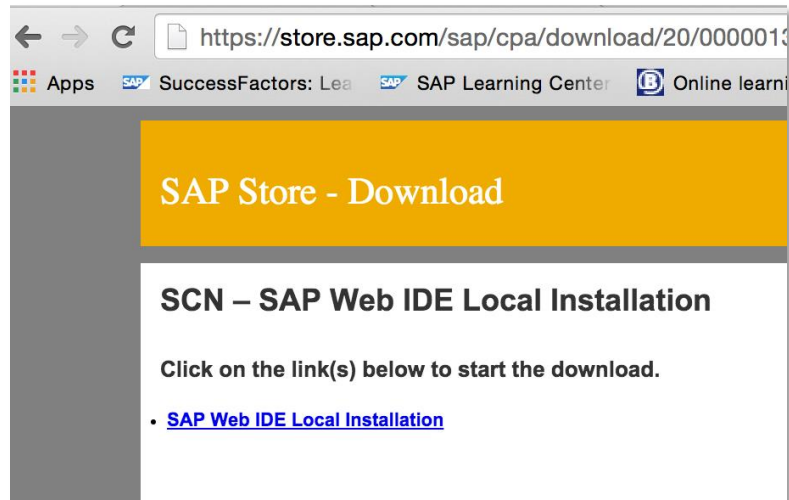
IDE Setup (cont.)

- Option 1: Cloud/SAP HCP
 - ♦ Pros
 - No maintenance required, always latest version
 - No license cost (for SAP system connections)
 - Fiori apps can be hosted in the cloud
 - Generates Component-preload.js
 - ♦ Cons
 - Requires SAP HANA Cloud Connector
 - May be slower compared to local installation
- Register at: <https://account.hanatrial.ondemand.com/>



IDE Setup (cont.)

- Option 2: Local Web IDE installation
 - ♦ Pros
 - ▶ Installed locally on Windows Local
 - ▶ Free, no licensing cost
 - ♦ Cons:
 - ▶ No updates so far after SAP Web IDE 1.12



Warning

Do not use Fiori Toolkit for Eclipse, it is deprecated!

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Extending Fiori App

- Import original Fiori app

The diagram illustrates the process of importing a Fiori app into SAP Web IDE. It consists of three main steps:

- Step 1: SAP Web IDE Home Screen**
The SAP Web IDE interface shows the 'Import an Application' section. The 'SAPUI5 ABAP Repository' option is highlighted with a yellow box.
- Step 2: Select Application from SAPUI5 ABAP Repository**
A dialog box titled 'Select Application from SAPUI5 ABAP Repository' is shown. It displays a table of applications available in the repository. The application 'GRC_ACREQ_CREO' is selected.
- Step 3: SAP Web IDE File Explorer**
The SAP Web IDE interface shows the file explorer view. The project 'GRC_ACREQ_CREO' is selected, and its contents are displayed in the right pane.

Table of Applications from SAPUI5 ABAP Repository:

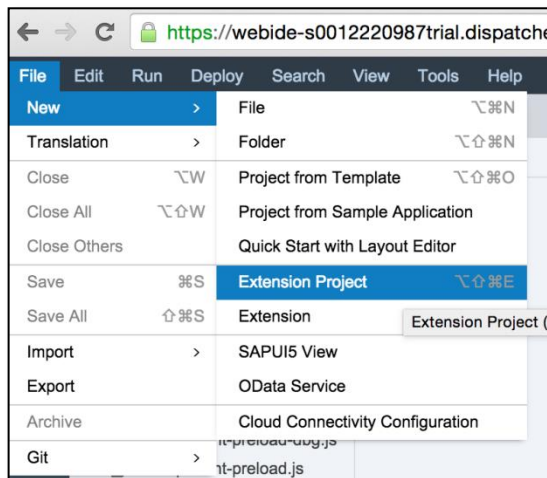
Application	Description
GRC_ACREQ_CREO	Access request on behalf of
GRC_ACREQ_MON	Check request status
GRC_AC_COMP_APV	Compliance Approval
GRC_SMT_RPT	GRC Smart Reports App
GRPC_CTRL_PERF	GRPC Control Performance
GRRM_UI5BOWTIE	Bow-tie
ILMRWC	Reporting work center - ILM Cockpit

File Explorer View (GRC_ACREQ_CREO):

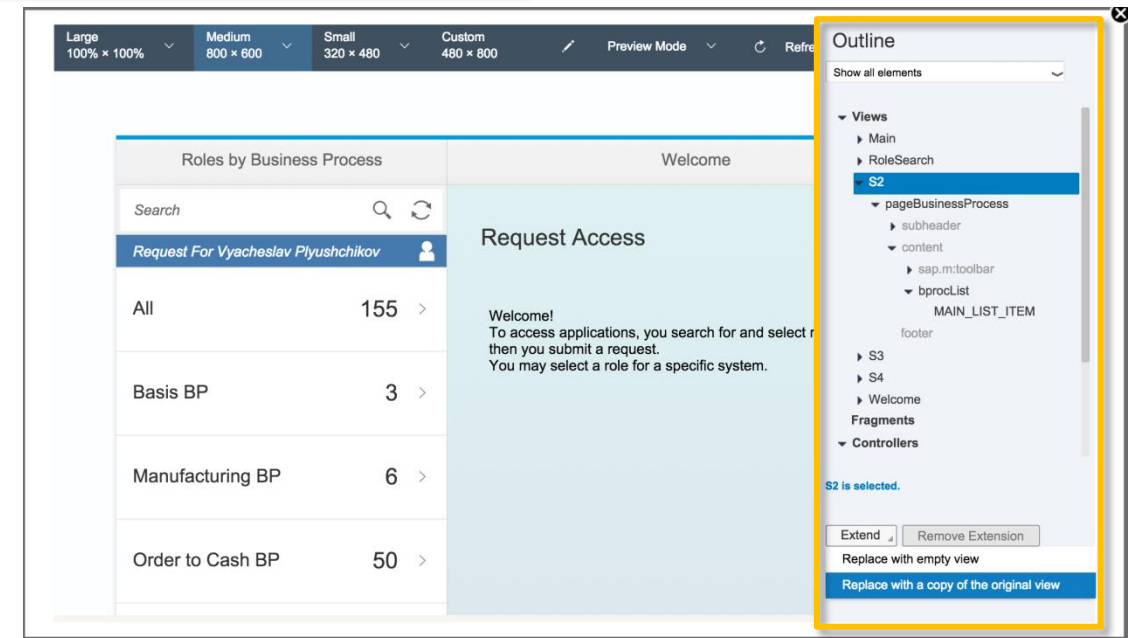
- Local
 - GRC_ACREQ_CREO
 - i18n
 - model
 - view
 - .project.json
 - .Ui5RepositoryBinaryFiles
 - .Ui5RepositoryIgnore
 - .Ui5RepositoryTextFiles
 - Component-dbg.js
 - Component-preload-dbg.js
 - Component-preload.js
 - Component.js
 - Configuration-dbg.js
 - Configuration.js
 - Main-dbg.controller.js
 - Main.controller.js
 - Main.view.xml
 - neo-app.json
 - version.json

Extending Fiori App (cont.)

- Create Extension project

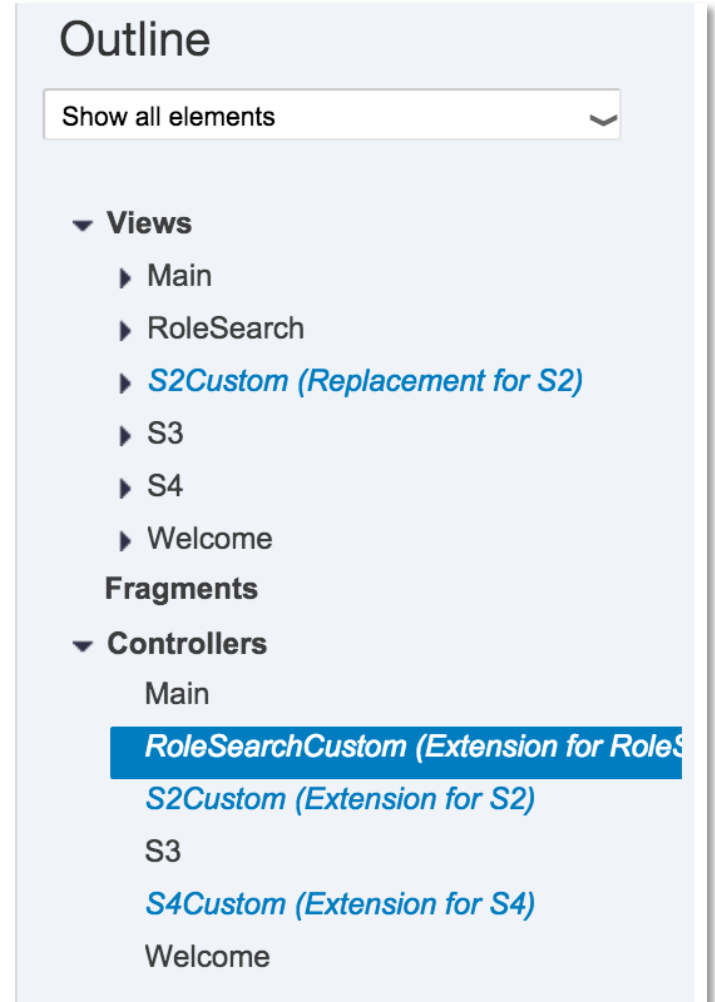


A screenshot of the 'NEW EXTENSION PROJECT' dialog box. The '1 Original Application and Name' step is active. The 'Original Application*' field contains '/GRC_ACREQ_CREO'. The 'Extension Project*' field contains 'GRC_ACREQ_CREOExtension'. There is a 'Select Application' button next to the 'Original Application*' field. A checkbox for 'Import original application' is present and unchecked.



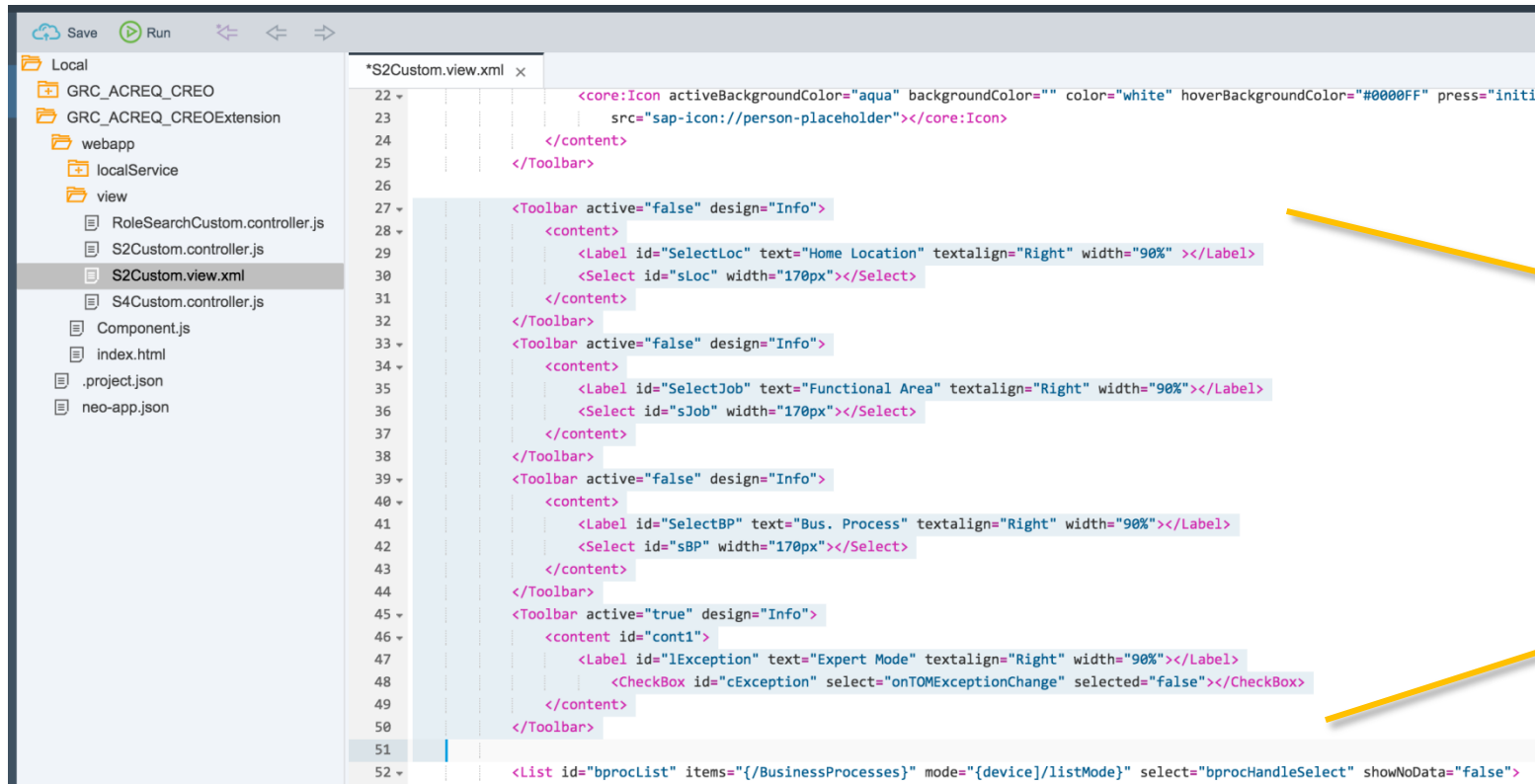
Extending Fiori App (cont.)

- Extend only views and controllers that need to be changed
- For Business Case we extended:
 - ♦ View S2: adding search filters
 - ♦ Controller S2: fill in search filters
 - ♦ Controller RoleSearch: send filter criteria to GRC
 - ♦ Controller S4: hide extra custom fields

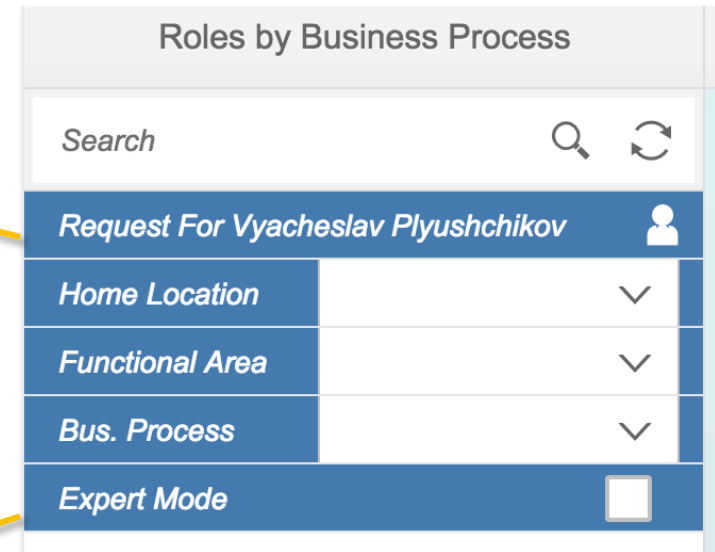


Extending Fiori App (cont.)

- Add Needed UI Elements to S2:



```
*S2Custom.view.xml x
22 <core:Icon activeBackgroundColor="aqua" backgroundColor="" color="white" hoverBackgroundColor="#0000FF" press="init
23 src="sap-icon://person-placeholder"></core:Icon>
24 </content>
25 </Toolbar>
26
27 <Toolbar active="false" design="Info">
28 <content>
29 <Label id="SelectLoc" text="Home Location" textalign="Right" width="90%"></Label>
30 <Select id="sLoc" width="170px"></Select>
31 </content>
32 </Toolbar>
33 <Toolbar active="false" design="Info">
34 <content>
35 <Label id="SelectJob" text="Functional Area" textalign="Right" width="90%"></Label>
36 <Select id="sJob" width="170px"></Select>
37 </content>
38 </Toolbar>
39 <Toolbar active="false" design="Info">
40 <content>
41 <Label id="SelectBP" text="Bus. Process" textalign="Right" width="90%"></Label>
42 <Select id="sBP" width="170px"></Select>
43 </content>
44 </Toolbar>
45 <Toolbar active="true" design="Info">
46 <content id="cont1">
47 <Label id="lException" text="Expert Mode" textalign="Right" width="90%"></Label>
48 <CheckBox id="cException" select="onTOMExceptionChange" selected="false"></CheckBox>
49 </content>
50 </Toolbar>
51
52 <List id="bprocList" items="{/BusinessProcesses}" mode="{device}/listMode" select="bprocHandleSelect" showNoData="false">
```



Roles by Business Process	
Search	
Request For Vyacheslav Plyushchikov	
Home Location	▼
Functional Area	▼
Bus. Process	▼
Expert Mode	<input type="checkbox"/>

Extending Fiori App (cont.)

- Add logic to controller S2:

The screenshot shows the SAP Fiori development environment. On the left, the project explorer displays the file structure: Local > GRC_ACREQ_CREO > GRC_ACREQ_CREOExtension > webapp > localService > view > S2Custom.controller.js. The main editor displays the code for S2Custom.controller.js. The code defines three OData models: mModelLOC, mModelJOB, and mModelBP. mModelLOC is an ODataModel for the path '/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV'. mModelJOB is an ODataModel for the path '/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV'. mModelBP is an ODataModel for the path '/sap/opu/odata/sap/GRC_ACCREQ_CREATE_OTHER'. The code also defines three templates: loctemplate, sjob, and bptemplate. loctemplate is a core.Item with key '{Value}' and text '{Text}'. sjob is a core.Item with key '{BusinessProcessID}' and text '{Description}'. bptemplate is a core.Item with key '{BusinessProcessID}' and text '{Description}'. The code uses sap.ui.getCore().setModel() to set the models and sap.ui.getCore().getModel() to get the models. The code also uses sap.ui.getCore().getTemplate() to get the templates. The code uses sap.ui.getCore().getTemplate() to get the templates. The code uses sap.ui.getCore().getTemplate() to get the templates.

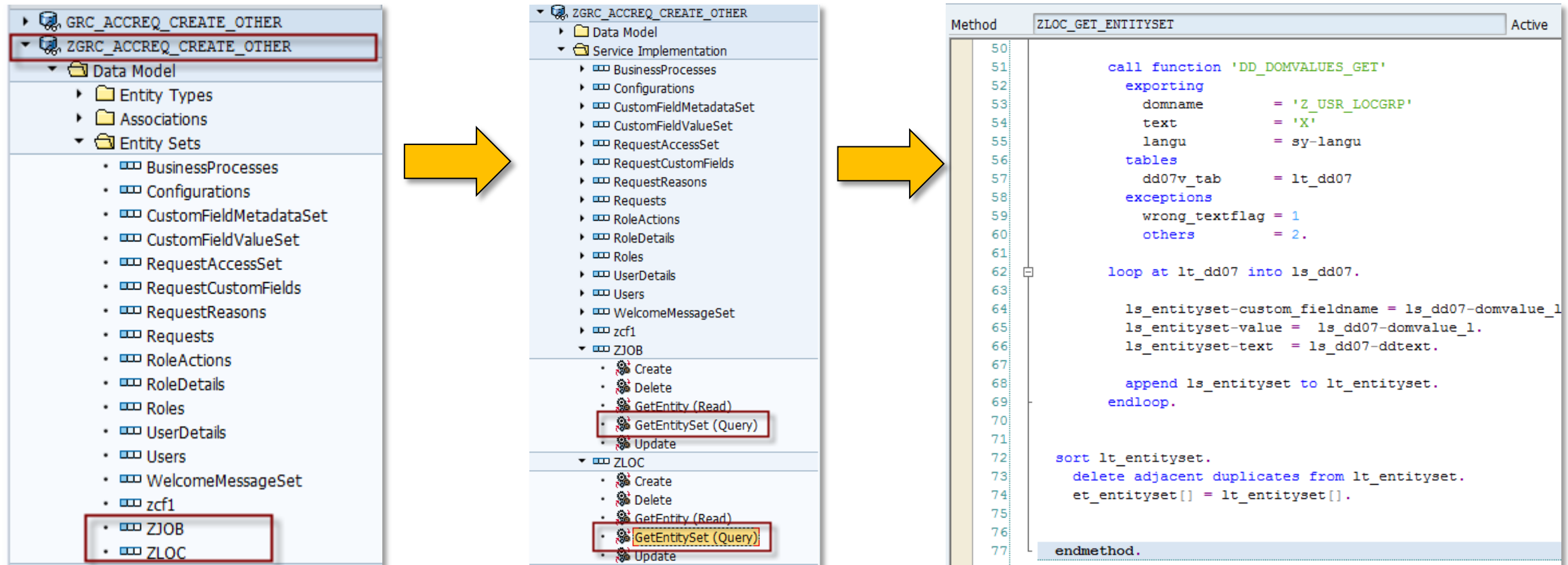
```
57
58     var mModelLOC = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV");
59     sap.ui.getCore().setModel(mModelLOC, "data_model");
60     var loctemplate = new sap.ui.core.Item({
61         key: "{Value}",
62         text: "{Text}"
63     });
64     var sloc = this.byId("sLoc");
65     sloc.setModel(sap.ui.getCore().getModel("data_model"));
66     sloc.bindAggregation("items", {
67         path: "/ZLOC",
68         length: 300,
69         template: loctemplate
70     });
71
72     var mModelJOB = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV");
73     sap.ui.getCore().setModel(mModelJOB, "jdata_model");
74     var sjob = this.byId("sJob");
75     sjob.setModel(sap.ui.getCore().getModel("jdata_model"));
76     sjob.bindAggregation("items", {
77         path: "/ZJOB",
78         length: 20,
79         template: loctemplate
80     });
81
82     var mModelBP = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/GRC_ACCREQ_CREATE_OTHER");
83     sap.ui.getCore().setModel(mModelBP, "jdata_model");
84     var bptemplate = new sap.ui.core.Item({
85         key: "{BusinessProcessID}",
86         text: "{Description}"
87     });
```

Build custom “Z” OData Services where required

... or reuse standard OData Services

Extending OData Services

- Transaction SEGW
 - Copy original OData Service and extend the copy



Extending OData Services (cont.)

- Transaction /IWFND/MAINT_SERVICE
 - ♦ Register OData service, maintain ICF node and system alias

The screenshot shows the SAP configuration interface for ICF nodes and system aliases. The top bar displays the transaction path: BEP > ZGRC ACCREQ CREATE OTHER SRV > 1 > Access Request Creation > ZGRC ACCREQ CREATE OTHER S. Below this, there are tabs for 'ICF Node', 'Call Browser', and 'SAP Gateway Client'. The 'ICF Nodes' table has columns for Status, ICF Node, Session Time-out, Soft State, and Description. It contains one entry: ODATA with a status of 'OK' and a session time-out of '00:00:00'. To the right, the 'System Aliases' section includes buttons for 'Add System Alias', 'Remove System Alias', and 'Customizing'. Below these buttons is a table with columns for 'SAP System Alias' and 'Description', containing one entry: LOCAL with the description 'Local System Alias'.

Status	ICF Node	Session Time-out	Soft State	Description
OK	ODATA	00:00:00		Standard M

SAP System Alias	Description
LOCAL	Local System Alias



The screenshot shows a user interface titled 'Roles by Business Process'. It features a search bar at the top with a magnifying glass icon and a refresh icon. Below the search bar, the user's name 'Request For Vyacheslav Plyushchikov' is displayed next to a user icon. The interface has several dropdown menus: 'Home Location' (set to '<not selected>'), 'Functional Area' (set to '<not selected>'), 'Bus. Process' (set to 'Algeria'), and 'Expert Mode' (set to 'Argentina'). A list of countries is visible on the right side of the screen, including Argentina, Australia, Austria, Brazil, Canada, Chile, and USA.

Home Location	Functional Area	Bus. Process	Expert Mode
<not selected>	<not selected>	Algeria	Argentina

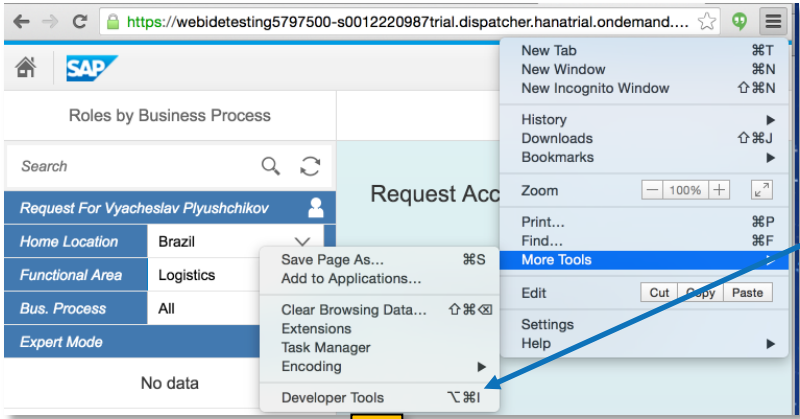
- Argentina
- Australia
- Austria
- Brazil
- Canada
- Chile
- USA

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Debugging in Web Browser

- Chrome is recommended for HTML5/Fiori development



Start developer tools in the browser

Inspect UI elements

- Automatic highlight
- Access to properties

Debugging in Web Browser (cont.)

The screenshot displays a web browser window with the SAP Fiori application 'Roles by Business Process'. The application shows a search filter for 'Request For Vyacheslav Plyushchikov' and a table with columns 'Home Location', 'Functional Area', 'Bus. Process', and 'Expert Mode'. The table is currently empty, showing 'No data'.

The browser's developer tools are open, showing the 'Sources' panel. The file 'S2Custom.controller.js' is selected, and the code is paused at line 636, which is highlighted in blue. A callout box labeled 'Inspect source code' points to this line. The code at line 636 is:

```
636 if (this.byId("sLoc").getSelectedKey() === "*") {
```

The 'Call Stack' panel on the right shows the current call stack, with the top entry being 'S2Custom.controller.js:636 sap.ui.controller.applyBackendSearchPattern'. Below the call stack, the 'Scope' panel shows the local variables: 'b: L.extend.constructor', 'f: "*" (this)", and 'zLocJobm: undefined'. The 'Breakpoints' panel shows a breakpoint set at line 636 of 'S2Custom.controller.js'.

A callout box labeled 'Debug, set breakpoints, view variables, control execution flow' points to the 'Paused in debugger' button in the browser's top toolbar.

Tracing OData Calls

- User SAP Gateway tools
 - ♦ **/IWFND/ERROR_LOG**
 - Gateway error log: records errors for calls to SAP GW
 - ♦ **/IWFND/TRACES**
 - Gateway call traces: performance and payload traces
 - ♦ **/IWFND/GW_CLIENT**
 - SAP Gateway Client: analyze and repeat calls

Gateway Error Log

- First log to check if calls are failing

SAP Gateway: Error Log

Re-Select

Overview

Line	Entr...	Date	Time	T100 Error ID	T10...	Error ...	ICF Node	B	Error Text
3	1	12.01.2016	17:49:44	/IWFND/MED170		1		<input type="checkbox"/>	No service found for namespace , name GRC_ACCREQ_CREATE_OTHER_SRV, version 0
2	1		17:49:27	/IWBEP/CM_MGW_RT021		1	odata	<input checked="" type="checkbox"/>	Method 'BUSINESSPROCESSE_GET_ENTITYSET' not implemented in data provider clas
1	1		17:49:01	/IWFND/CM_CONSUMER122		1	odata	<input type="checkbox"/>	Invalid system query options value

Error Context

Ex...	Name	Value
	..ERROR_CONTEXT	
	..ERROR_INFO	No service found for namespace , name GRC_ACCREQ_CREATE_OTHER_SRV, version 0001
	..ERROR_RESOLUTION	
	...SAP_NOTE	See SAP Note 1797736 for error analysis
	...LINK_TO_SAP_NOTE	https://service.sap.com/sap/support/notes/1797736

Review error log

Analyze the call,
replay if required

Gateway Traces

- Analyze OData calls
- Review performance trace
- Replay calls to troubleshoot/debug issues

SAP Gateway: Tracing Tools

Users & Request URI Prefix

- TST-GRC-USR2

Configuration Performance Trace **Payload Trace**

Request URI Today <-> All Traces

Payload Trace: Client 100 User TST-GRC-USR2

Service Call Info	Method	Proc. Time	Appl. Time	Req. Size	Resp. Size	Format	Date	Time	Expiry Date	Status
/sap/GRC_ACCREQ_CREATE_OTHER/BusinessProcesses?\$skip...	GET	350	281	0	4.490	json	16.01.2016	12:04:18	30.01.2016	■
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	131	31	0	1	txt	16.01.2016	12:04:18	30.01.2016	■
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	132	33	0	1	txt	16.01.2016	12:04:18	30.01.2016	■
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	146	0	0	953	xml	16.01.2016	12:04:17	30.01.2016	❌
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	149	34	0	1	txt	16.01.2016	12:04:17	30.01.2016	■
/sap/GRC_ACCREQ_CREATE_OTHER/BusinessProcesses?\$skip...	GET	364	283	0	4.490	json	16.01.2016	12:04:17	30.01.2016	■
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	172	0	0	953	xml	16.01.2016	12:04:17	30.01.2016	❌
/sap/GRC_ACCREQ_CREATE_OTHER/Users?\$filter=UserID%20...	GET	182	94	0	900	json	16.01.2016	12:04:06	30.01.2016	■
/sap/GRC_ACCREQ_CREATE_OTHER/WelcomeMessageSet?\$e	GET	134	57	0	5.159	json	16.01.2016	12:04:01	30.01.2016	■

Gateway Client

- Retrieve OData services metadata
- Execute OData calls, review responses, troubleshoot issues

The screenshot displays the SAP Gateway Client application window. The title bar reads "SAP Gateway Client". The menu bar includes "Edit", "Goto", "Metadata", "System", and "Help". The toolbar contains various icons for file operations and execution. The main interface is divided into several sections:

- Toolbar:** Includes buttons for "Execute", "Select", "Maintain Service", "Service Implementation", "EntitySets", and "Add URI Option".
- HTTP Method:** Radio buttons for GET (selected), POST, PUT, PATCH, MERGE, and DELETE.
- Request URI:** A text field containing "/sap/opu/odata/sap/GRC_ACCREQ_CREATE_OTHER/?\$format=xml".
- Protocol:** Radio buttons for HTTP (selected) and HTTPS.
- Test Group:** A text field.
- Test Case:** A text field.
- HTTP Request:** A section with a text area for the request body and buttons for "Add File", "Remove File", and "Data Explorer".
- HTTP Response:** A section showing the response headers and body. The headers table is as follows:

Header Name	Value
~status_code	200
~status_reason	OK

The response body is an XML document:

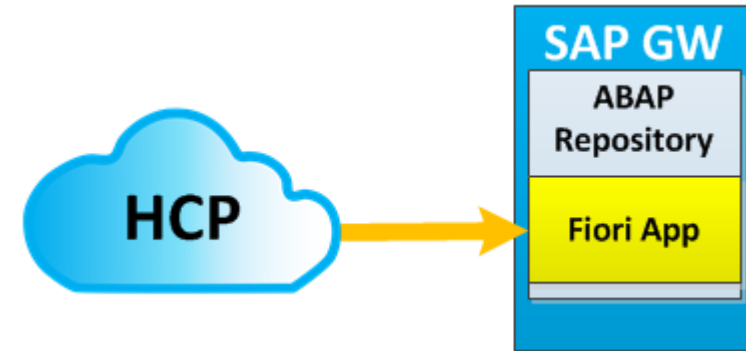
```
- <app:workspace>
  <atom:title type="text">Data</atom:title>
  - <app:collection sap:content-version="1" href="CustomFieldMetadataSet">
    <atom:title type="text">CustomFieldMetadataSet</atom:title>
    <sap:member-title>CustomFieldMetadata</sap:member-title>
  </app:collection>
  - <app:collection sap:content-version="1" href="Requests">
    <atom:title type="text">Requests</atom:title>
    <sap:member-title>Request</sap:member-title>
  </app:collection>
  - <app:collection sap:content-version="1" href="RequestAccessSet">
    <atom:title type="text">RequestAccessSet</atom:title>
```


What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Deploying Fiori App

- Deploy app code to ABAP system
 - ◊ Direct Deployment to SAPUI5 ABAP Repository
 - ◊ Export/Import to SAPUI5 ABAP Repository
- Give users access to Fiori app
 - ◊ Navigation Design
 - ◊ User Permissions



Direct Deployment to ABAP Repository

• Direct Deployment to SAPUI5 ABAP Repository

DEPLOY TO SAPUI5 ABAP REPOSITORY 1 Deployment Options > 2 Deploy a New Application > Next

1 Deployment Options

System * SAPSYSTEM

☒ Deploy a new application
☐ Update an existing application

DEPLOY TO SAPUI5 ABAP REPOSITORY 1 Deployment Options > 2 Deploy a New Application > Next

2 Deploy a New Application

Name * ZGRC_ACREQ_CREO

Description * ZGRC_ACREQ_CREO

Package * ZGRAC Browse

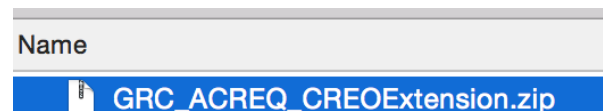
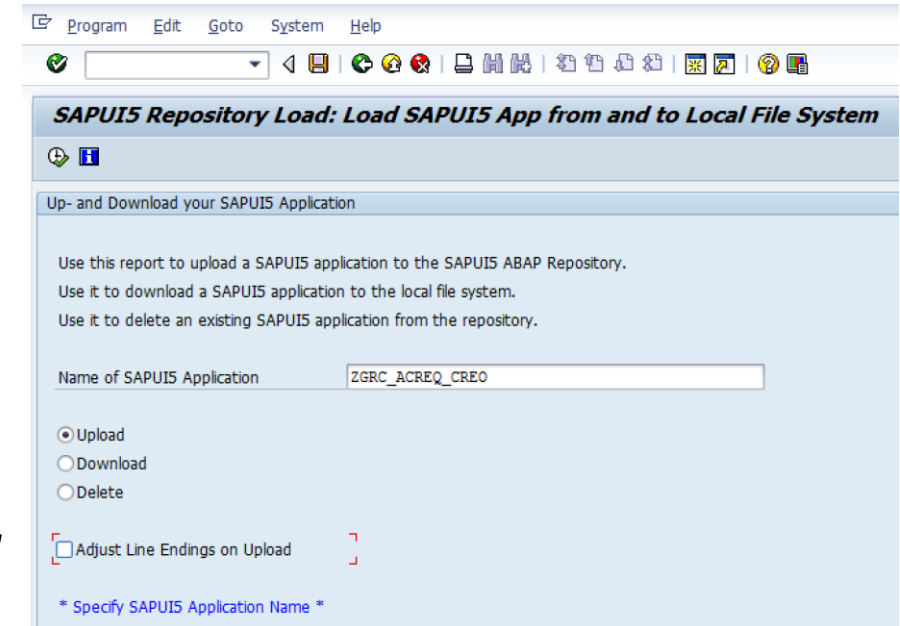
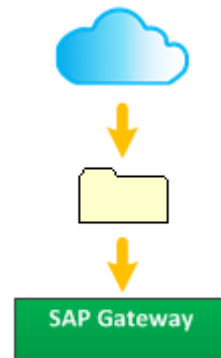
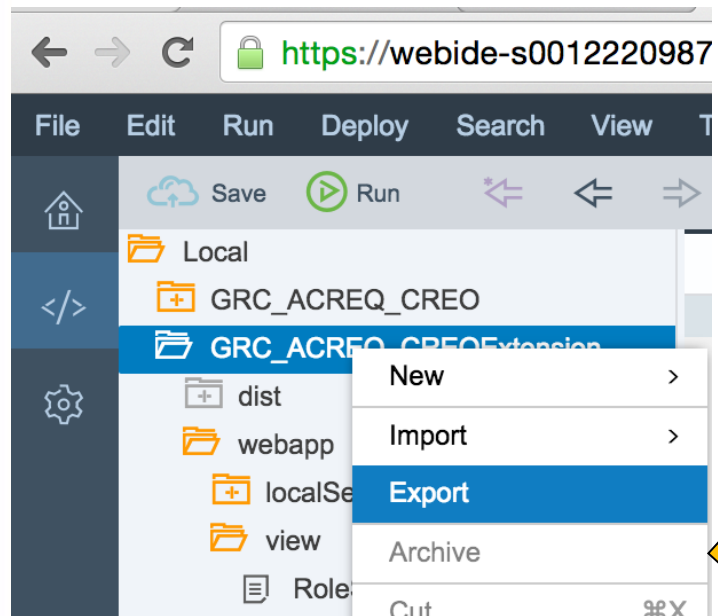
☒ Create a new request

Request Description New Fiori app

The Request ID is automatically generated

Export/Import to ABAP Repository

- Export from HCP
- Import via /UI5/UI5_REPOSITORY_LOAD report



Provide Access to Fiori App

- Define Semantic Object /UI2/SEM OBJ
- Define app alias in LPD_CUST



New Entries: Overview of Added Entries

Semantic Objects - Customer View

Semantic Object	Semantic Object Name	Semantic Object Description
ZAccessRequest	ZAccessRequest	Extended Access Request for Other

Change Launchpad - Role: UIGRC001 Instance: TRANSACTIONAL (EN)

☐ New Folder
 ☐ New Application
 ☐ Add Separator
 ☐ Delete
 ☐ Copy from Other Launchpad
 ☐ Link to a Repository Application

Customized Version

- GRC: Transactional Apps
 - Inactive Applications
 - Check Request Status
 - Request access
 - Access Request
 - Analyze Risks
 - Maintain Role
 - Analyze Users
 - Analyze Roles
 - Analyze Risks
 - Approve Access Request
 - Smart Reports
 - Request access for others
 - Z Request access for other**
 - Compliance approver

Link Details

Link Text: Z Request access for other

Description:

Application Type

Application Type: URL

Application Parameter

URL: /sap/bc/ui5_ui5/sap/zgrc_acreq_creo

System

System Alias:

Hide Advanced (Optional) Parameters

Advanced Parameters (Optional)

Application - Deactivation by User

Application Cannot be Removed from Launchpad: ☐

Application-Related Parameters

Application Alias: ZACCESSREQUESTO

Target App. Parameters:

Suspend/Resume: ☐

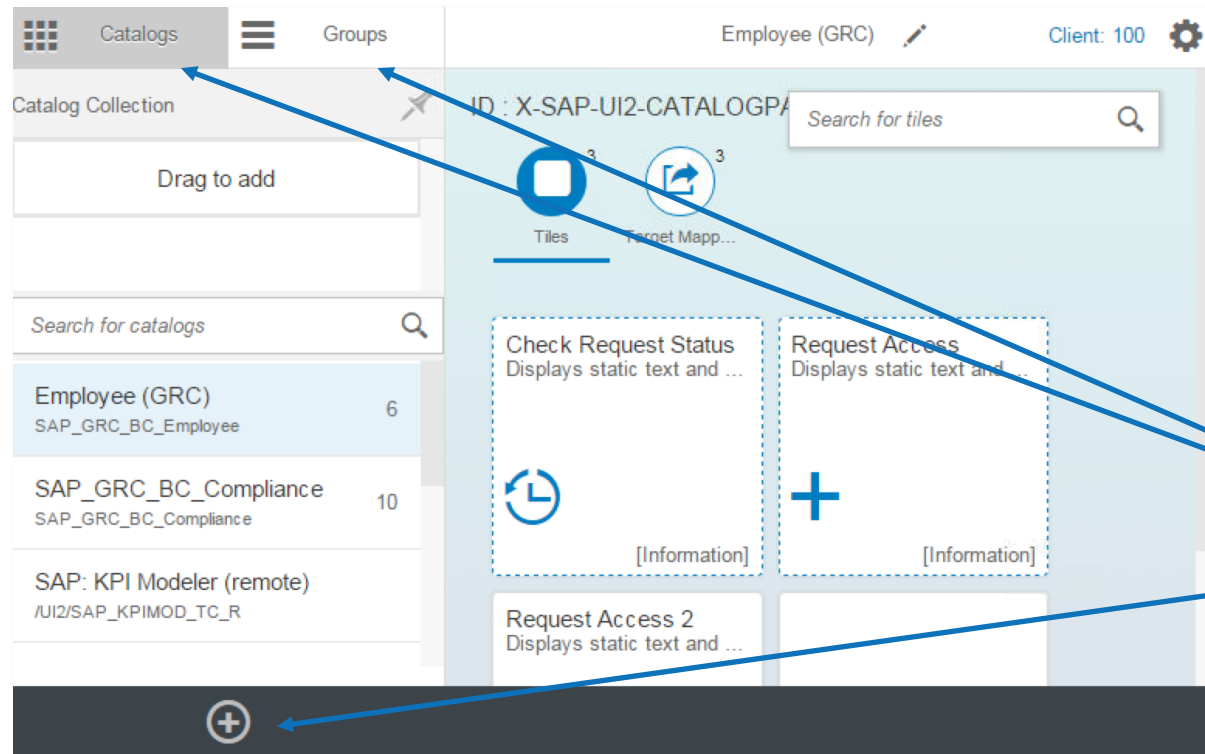
Proxy Class:

Additional Information: SAPUI5.Component=fcg.grc.acreq.creat...

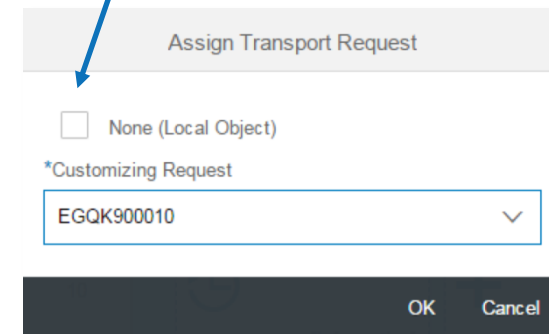
FPM Event ID:

Provide Access to Fiori App (cont.)

- Define Catalogs and Groups
 - ♦ /sap/bc/ui5_ui5/sap/arsrvc_upb_admn/main.htm
- Create tiles, target mappings, add tiles to groups



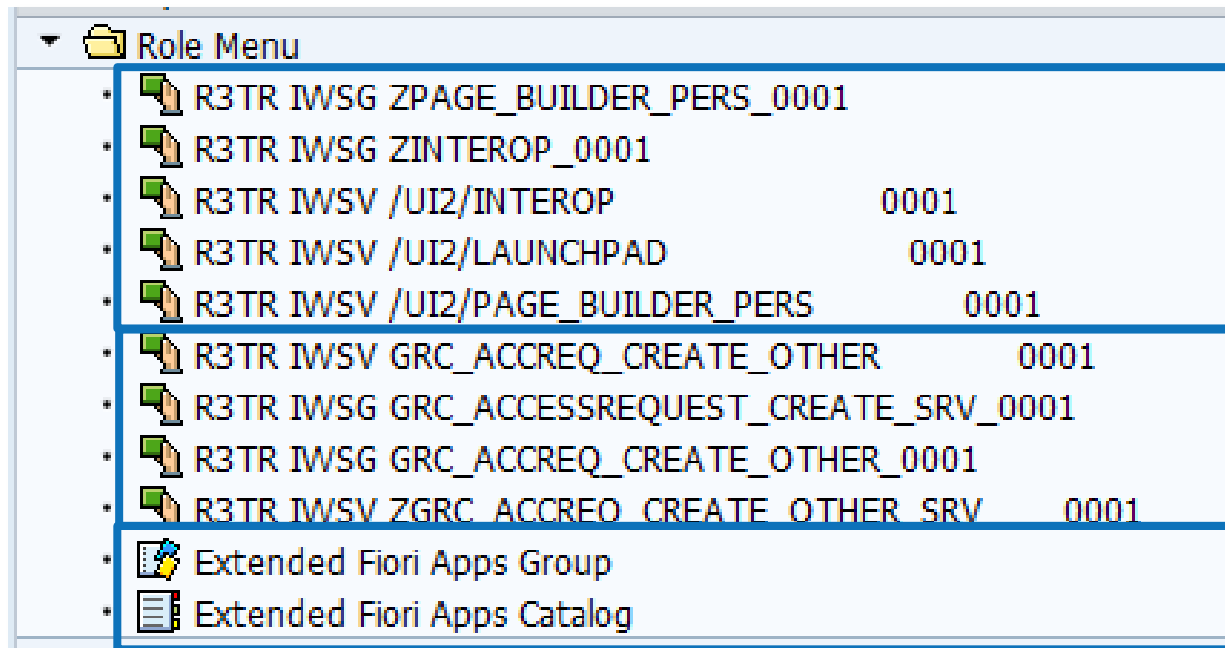
Make sure you assign transport request before configuration



Add Catalogs, Groups as needed

Provide Access to Fiori App (cont.)

- Define PFCG roles for Fiori Launchpad
 - ♦ Define access to catalogs, groups, OData services, required Fiori Launchpad services



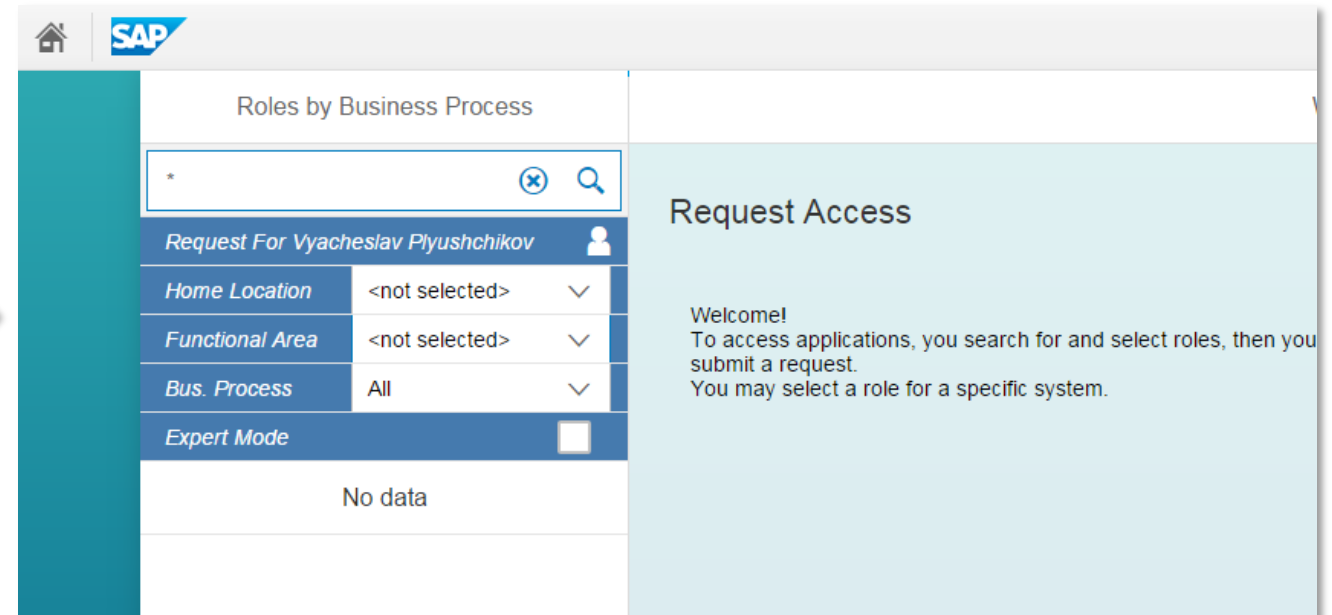
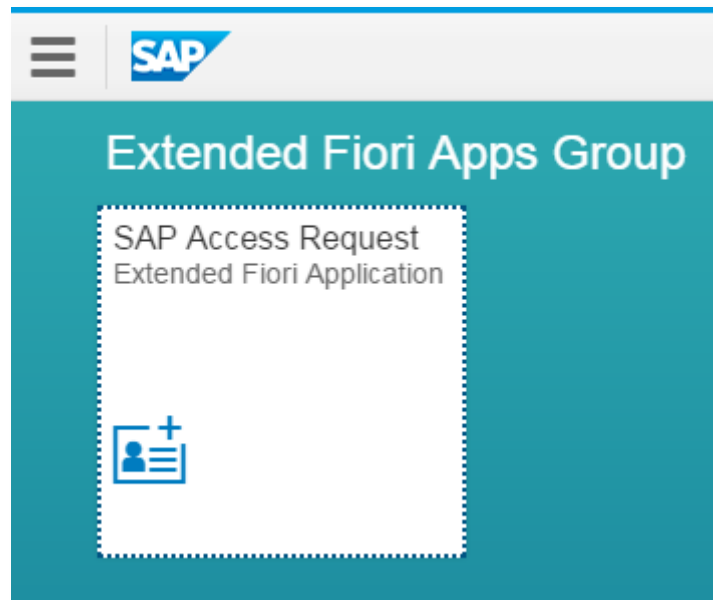
SAP UI authorization items

Fiori app items

Fiori Launchpad items

Provide Access to Fiori App (cont.)

- Assign new role to users
- User can see new tile in Fiori Launchpad



Demo: Modifying and Deploying Fiori App

SAP Web IDE in Action



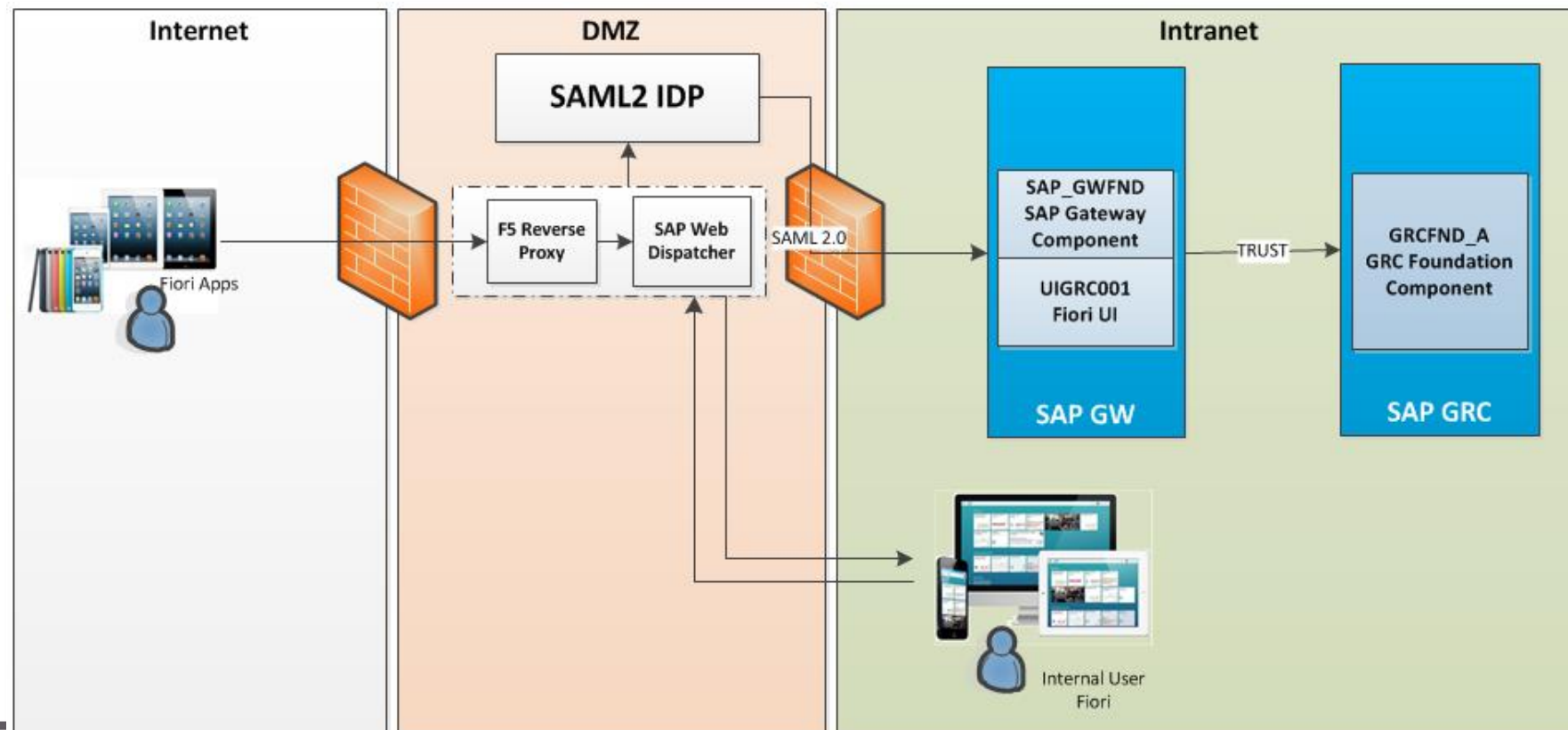
Demo

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

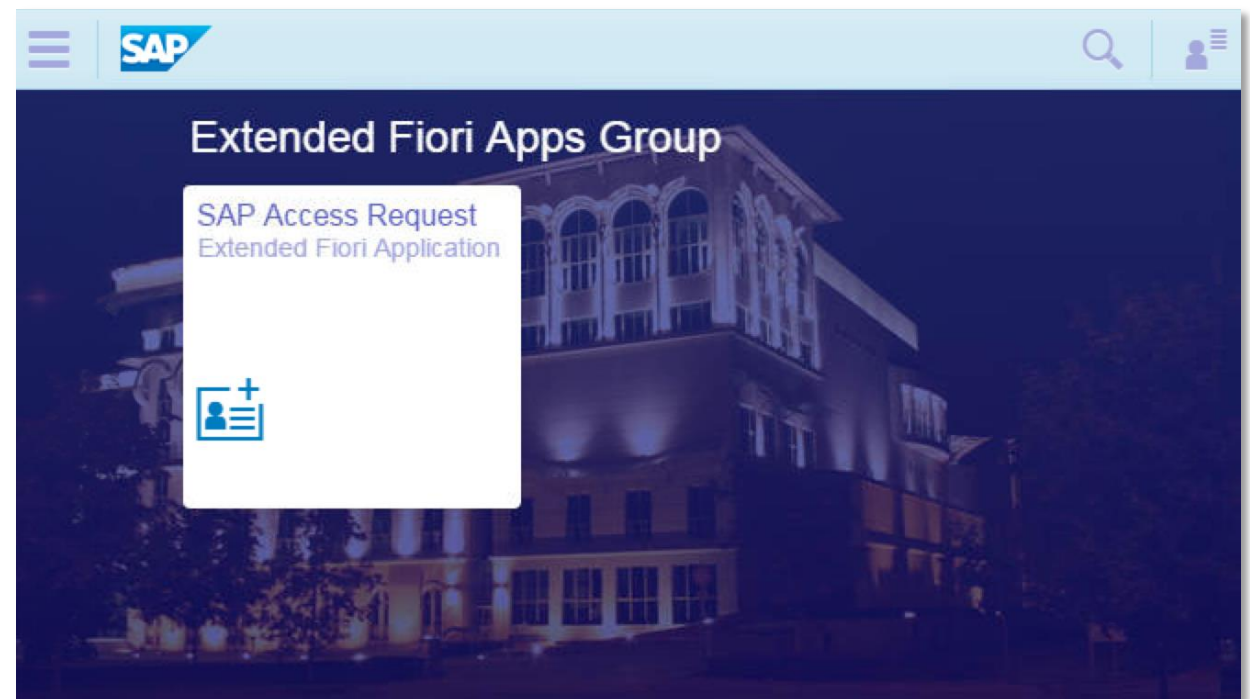
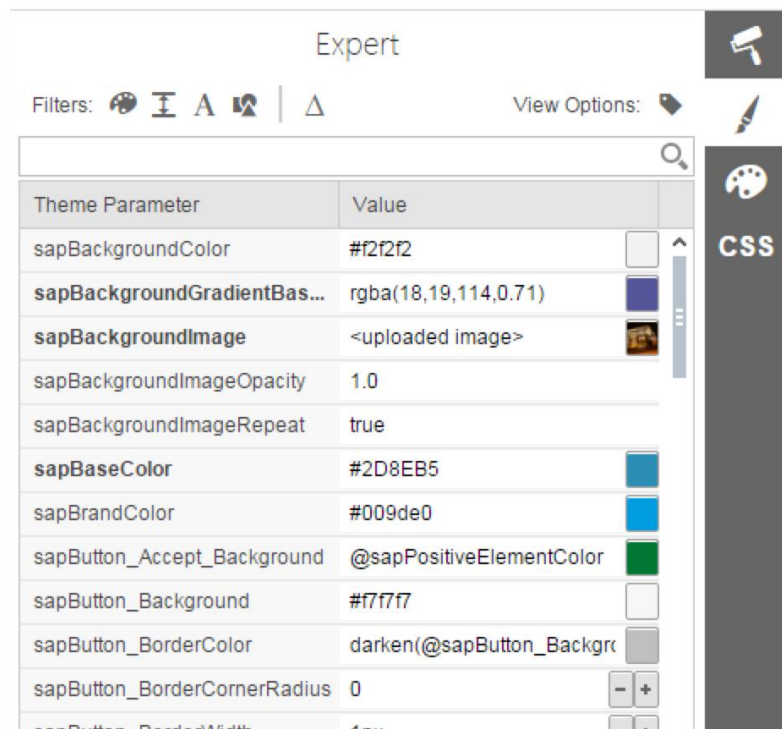
Single Sign-On

- Most common solution is based on SAML2
 - ♦ Uses HTTPS ports
 - ♦ Supports identity federation



Branding

- Theme Designer
 - ♦ /ui5/theme_designer
 - ♦ Copy existing theme to a new name and customize



What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Where to Find More Information

- <http://help.sap.com/grc-ac#section6>
 - ♦ GRC Access Control: SAP Fiori apps documentation on the SAP Help Portal
- <https://account.hanatrial.ondemand.com>
 - ♦ SAP HANA Cloud Platform: registration, access to management cockpit
- <https://sapui5.hana.ondemand.com>
 - ♦ SAP UI Development Toolkit: SAPUI5/Fiori development
- <http://help.sap.com/nwgateway#section7>
 - ♦ SAP Gateway 2.0: SAP GW/OData services development on the SAP Help Portal
- <http://advanced-view.com>
 - ♦ Vyacheslav Plyushchikov's blogs on GRC Access Control and Fiori apps

7 Key Points to Take Home

- SAP Fiori apps for GRC AC can easily be extended
- SAP provides free HCP-based IDE for developing and extending Fiori apps
- Use design thinking approach: start from “better future” goal, not from analyzing current problems
- Reuse as much as possible, change only app parts, where new functionality is needed
- Keep the app “lightweight” to load faster on mobile devices, no big images or large data pre-loads
- Build Fiori apps for access from the Internet: secure it with HTTPS, web filters, identity federation authentication
- Use trace and debug tools for front-end and back-end issue resolution

Your Turn!



Questions?

How to contact me:
Vyacheslav (Slava) Plyushchikov
slava.plyushchikov@gmail.com

Please remember to complete your session evaluation

Disclaimer

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. Wellesley Information Services is neither owned nor controlled by SAP SE.



Wellesley Information Services, 20 Carematrix Drive, Dedham, MA 02026
Copyright © 2016 Wellesley Information Services. All rights reserved.